UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

FRED B. DUFRESNE, Plaintiff and Counter-Defendant, v. MICROSOFT CORPORATION, ADOBE SYSTEMS INCORPORATED and MACROMEDIA, INCORPORATED, Defendants and Counter-Claimants.	Civil Action No. 02-11778 WGY

MICROSOFT'S MEMORANDUM IN SUPPORT OF ITS MOTION FOR SUMMARY JUDGMENT OF NON-INFRINGEMENT

TABLE OF CONTENTS

I. INTRODUCTION 4

II. BACKGROUND 4

A. ASP Processing 4

B. ASP Processing Example 7

III. CLAIM CONSTRUCTION 9

IV. LEGAL STANDARDS 10

A. Summary Judgment 10

B. Non-Infringement 10

V. ARGUMENT 10

A. ASP does not infringe the '712 patent because ASP does not include "executable tags." 11

B. ASP also does not infringe the '712 patent because "processing the source . . . by executing the tags" (claims 9 and 14) and "processing the source . . ., the process executing the tags" (claim 26) are absent in ASP. 11

C. DuFresne cannot rely on the doctrine of equivalents because DuFresne has produced no expert evidence on the subject and has been precluded from doing so by the Court. 13

D. DuFresne additionally cannot rely on the doctrine of equivalents because of prosecution history estoppel. 13

VI. CONCLUSION 15

TABLE OF AUTHORITIES

Cases

Anderson v. Liberty Lobby, Inc.,
477 U.S. 242 (1986) 10
AquaTex Indus., Inc. v. Techniche Solutions
479 F.3d 1320 (Fed. Cir. 2007) 13
Bayer AG v. Elan Pharm. Research Corp.
212 F.3d 1241 (Fed. Cir. 2000) 10, 14
Canton Bio. Med. v. Integrated Linear Techs., Inc.
216 F.3d 1367 (Fed. Cir. 2000) 10
Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co., Ltd.
535 U.S. 722 (2002) 14
Int'l Strategies Group, Ltd. v. Greenberg Traurig, LLP
482 F.3d 1 (1st Cir. 2007) 10
Kahn v. General Motors Corp.
135 F.3d 1472 (Fed. Cir. 1998) 10
Network, LLC v. Centraal Corp.
242 F.3d 1347 (Fed. Cir. 2001) 10
Seal-Flex, Inc. v. Athletic Track & Court Constr.
98 F.3d 1318 (Fed. Cir. 1996) 10
Southwall Techs., Inc. v. Cardinal IG Co.
54 F.3d 1570 (Fed. Cir 1995) 14

Statutes

Fed. R. Civ. P. 56(c) 10

I. INTRODUCTION

The accused Microsoft products do not infringe the '712 patent as a matter of law. The material facts relevant to this motion – in particular the operation of the accused Microsoft Active Server Page ("ASP") technology as laid out in the expert report of Roy Fielding – have never been disputed by DuFresne. Application of the Court's claim construction to this undisputed operation of Microsoft's ASP technology yields but one result: no reasonable jury could find that Microsoft's ASP technology infringes the '712 patent.

II. BACKGROUND[1]

ASP is a technology that can be used to provide static or dynamic web pages. [Declaration of Roy T. Fielding in Support of Microsoft Corporation's Motions for Summary Judgment ("Fielding Decl.") Ex. 1 at ¶¶ 19(e) & 57.] What sets ASP apart from other approaches is its versatility. [*Id*. at ¶¶ 25 & 26.] Specifically, an ASP file can include blocks of code written in a general-purpose programming language. [*Id*. at ¶ 19(c).] Computer programmers can use these code blocks to program functionality into an ASP file as they see fit. [*Id.* at ¶¶ 25 & 26.] Other technologies, by contrast, tip in favor of providing ease of implementation over versatility. [*Id*. at ¶ 39(a)(xiii).] They do so by requiring the use of special-purpose tags whose functionality is predetermined. [*Id*.] The '712 patent is an example of such an approach. [*Id*.] Because the ASP approach is different, it is accomplished through the use of particular file processing that differs from the '712 patent claims as construed by the Court.

A. ASP Processing

There are generally two steps involved in processing an ASP file written by a computer programmer. The first step is a parsing step, and the second step is a processing step. [Fielding

[1] The background section presents both material undisputed facts which are necessary to this motion and certain non-material, contextual facts. The material undisputed facts are indicated by a citation to Microsoft's statement of material undisputed facts. The contextual facts are indicated by a citation to the declaration of Roy Fielding.

Decl., Ex. 1 at ¶ 23(e)-(f).] These steps are discussed in more detail below with reference to Table 1:[2]

TABLE 1		
Stage 1: The Original ASP File	Stage 2: The Intermediate File (after the parsing step)	Stage 3: The HTML File (after the processing step)
<html tag> <%[CODE BLOCK]%> <html tag>	printf("<html tag>") [CODE BLOCK] Printf("<html tag>")	<html tag> html tag(s) resulting from [CODE BLOCK] <html tag>

Stage 1: The ASP File

The ASP file, as shown in the left column of Table 1, is the file written by the computer programmer. [*Id*. at ¶ 19(b).] ASP allows code blocks written in a general-purpose programming language (shown generically in blue) to be interspersed with literal text, such as HTML tags (shown in green), in a single ASP file. [*Id*. at ¶ 19.] The code blocks written in this general-purpose programming language are contained within the delimiters "<%", indicating the beginning of the code block, and "%>", indicating the end of the code block (shown in red). [*Id.* at ¶ 19(c).] Through the use of these delimiters, ASP permits code blocks to reside in the same file as conventional HTML tags and other text. [*Id*. at ¶ 19.]

Stage 2: The Intermediate File

When a client computer requests an ASP file from a web server, the web server invokes the ASP software component. [*Id.* at ¶ 23(d).] The ASP component parses the ASP file in order

[2] The tables in this memorandum have been reproduced in an appendix for ease of reference. The appendix can be found at the end of this memorandum.

to create a new file, denoted above as the intermediate file.[3] [Microsoft's Statement Of Material Facts On Which There Is No Genuine Issue to Be Tried ("MS MUF") at ¶ 1.] This intermediate file consists solely of general-purpose programming language statements.[4] [Fielding Decl., Ex. 1 at ¶ 23(e).] The intermediate file is shown in the middle column of Table 1.

For each literal text block in the ASP file (i.e., anything outside of the delimiters "<%" and "%>"), that literal text (shown in green) is placed into a general-purpose print command (shown in black). [*Id.* at ¶ 23(e)(i).] On the other hand, when the ASP component encounters the beginning and ending ASP delimiters "<%" and "%>", those delimiters are deleted, and the code block (shown in blue) between the delimiters is copied verbatim to the intermediate file. [MS MUF at ¶¶ 2 & 3.] Thus, the delimiters "<%" and "%>" serve as a signal to the ASP component to not do anything to the contents of the code block during parsing. [Fielding Decl., Ex. 1 at ¶ 23(e)(vi).] With the delimiters "<%" and "%>" excluded from the file, only the copied code block remains. [*Id.* at ¶ 23(e)(v).] The ASP file itself is not altered during the parsing process. [*Id*. at ¶ 23(e)(v).]

There are advantages to parsing an ASP file into in an intermediate file consisting solely of general-purpose programming language statements. Because web servers can retain the intermediate file generated in response to an ASP file request, and use it for subsequent requests, the time required to parse the ASP file subsequently is saved. [*Id.*] The previously generated intermediate file is simply reused for later requests. [*Id.*]

---

[3] There have been two major releases of ASP. They are ASP ("ASP Classic") and ASP.NET ("ASP.NET"). Both ASP Classic and ASP.NET parse the ASP file to generate an intermediate file. In ASP.NET, the intermediate file is additionally compiled by a general-purpose programming language compiler prior to being processed into an HTML file, as described in Stage 3. In other words, in ASP.NET, there are four stages rather than three. [Fielding Decl., Ex. 1 at ¶ 23(f).] Thus, ASP.NET is even further removed than ASP Classic from the '712 patent.

[4] "printf('<html tag>')" is a programming statement, where "printf" is a programming command and "<html tag>" is the argument for that command (i.e., it is what gets printed).

The intermediate file, in syntax and structure, looks like a script written using a general purpose programming language, such as a CGI script written using Perl. [*Id.* at ¶ 23.] It is noteworthy that DuFresne explicitly distinguished his invention from prior art CGI scripts in his patent specification: "CGI scripts, however, are complex and difficult to program. Each script requires customization to implement a particular Web application in a particular way… [T]he present invention provides dynamic Web environment without the complexities associated with the CGI programming." [Declaration of Jennifer K. Bush in Support of Microsoft's Motion for Summary Judgment of Non-Infringement ("Bush Decl."), Ex. A at 8:38-50.] He distinguished other types of scripts as well during prosecution. [Bush Decl., Ex. C at 10 ("Lyons, on the other hand, discloses a script-based data communications system.").]

Stage 3: The HTML File

After parsing, the ASP component processes the intermediate file to generate a third file, the HTML file, shown in the right column of Table 1. [MS MUF at ¶ 4.] The general-purpose programming language statements in the intermediate source are processed into literal text (shown in green and blue). [Fielding Decl., Ex. 1 at ¶ 23(f)(i).] The resulting HTML file is then forwarded to the client computer that requested the ASP file. [MS MUF at ¶ 5.]

B. ASP Processing Example

Below is an example of the general ASP processing approach described above.

TABLE 2		
Stage 1: ASP File	Stage 2: Intermediate File (after the parsing step)	Stage 3: HTML File (after the processing step)
<html> <body> <% For i=1 To 5 printf" Microsoft " Next %> </body> </html>	printf("<html>") printf("<body>") For i=1 To 5 printf(" Microsoft ") Next printf("</body>") printf("</html>")	<html> <body> Microsoft Microsoft Microsoft Microsoft Microsoft </body> </html>

Stage 1: The ASP File

The ASP file, shown in the left column of Table 2, includes both HTML tags (shown in green) and a code block (shown in blue). [Fielding Decl., Ex. 1 at ¶ 19.] The code block is delimited by the "<%" and "%>" ASP delimiters (shown in red), which indicate the start and end of the code block respectively. [*Id.* at ¶ 19(c).] In this example, the function of the code block is to simply print " Microsoft " five times.[5] [*Id.* at ¶ 23(f)(i).]

Stage 2: The Intermediate File

After the web server receives a request for the ASP file, the ASP component parses the ASP file to create the intermediate file shown in the middle column of Table 2. [MS MUF at ¶ 1.] During parsing, each of the HTML tags (shown in green) is placed into a general-purpose print command (shown in black). [Fielding Decl., Ex. 1 at ¶ 23(e)(i).] The code block that prints " Microsoft " five times (shown in blue) is copied verbatim into the intermediate file because it was placed between the ASP delimiters "<%" and "%>". [MS MUF at ¶ 2.] Having served their purpose, these delimiters are excluded from the intermediate file. [*Id.* at ¶ 3.] Thus,

[5] This ASP file generates a static web page because every time the code block is processed, it prints " Microsoft " five times. The code block, however, is versatile and could be written to create any kind of content, including dynamic content.

the intermediate file consists entirely of general-purpose programming language code. [Fielding Decl., Ex. 1 at ¶ 23(e).]

Stage 3: The HTML File

The processing of the intermediate file results in the HTML content shown in the right column of Table 2. [MS MUF at ¶ 4.] All the general-purpose programming language statements in the intermediate file, including the code block, have been processed (shown in green and blue). [Fielding Decl., Ex. 1 at ¶ 23(f).] This new HTML file is forwarded by the server back to the client. [MS MUF at ¶ 5.]

The operation of ASP, as generally described above, is not in dispute. This process was described by Roy T. Fielding, Microsoft's technical expert, in his Expert Report on Non-Infringement dated November 28, 2003. None of the material facts described above were generally disputed by David Klausner, DuFresne's expert, in his Supplemental Report Regarding Infringement dated December 19, 2003.

III. CLAIM CONSTRUCTION

During the Markman hearing on May 21, 2007, the Court issued tentative constructions for each of the disputed claim terms. The following constructions are pertinent to this analysis:

Claim Term	Claims	The Court's Tentative Construction
executable tag	9, 14, 26	Delimited computer codes that are replaced with static and/or dynamic values upon execution of the tag
processing the source … by executing the tags processing the source …, the process executing the executable tags	9, 14 26	Fully processing the hypertext source into which executable tags were inserted, by replacing each executable tag in the source with a corresponding value and expanding any resulting dynamic values until they are static.
hypertext source to a displayable page	9, 14, 26	Predefined text with codes which indicate how a page should be displayed by a browser

IV. LEGAL STANDARDS

A. Summary Judgment

Summary judgment is appropriate when there is no genuine issue as to any material fact and the moving party is entitled to judgment as a matter of law. Fed. R. Civ. P. 56(c); *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 251-52 (1986); *Seal-Flex, Inc. v. Athletic Track & Court Constr.*, 98 F.3d 1318, 1321 (Fed. Cir. 1996); *Int'l Strategies Group, Ltd. v. Greenberg Traurig, LLP*, 482 F.3d 1, 6 (1st Cir. 2007).

B. Non-Infringement

Summary judgment on a question of patent infringement is appropriate "when no material fact is in dispute, or when no reasonable trier of fact could find facts whereby the nonmoving party could prevail." *Canton Bio. Med. v. Integrated Linear Techs., Inc.*, 216 F.3d 1367, 1369 (Fed. Cir. 2000). The plaintiff bears the burden of proving infringement by a preponderance of the evidence. *Kahn v. General Motors Corp.*, 135 F.3d 1472, 1476 (Fed. Cir. 1998). For literal infringement, every single claim limitation must be literally present in the accused product. *Network, LLC v. Centraal Corp.*, 242 F.3d 1347, 1353 (Fed. Cir. 2001); *Bayer AG v. Elan Pharm. Research Corp.*, 212 F.3d 1241, 1247 (Fed. Cir. 2000).

V. ARGUMENT

In this case, as a matter of law and undisputed fact, ASP, and methods of using ASP, do not directly infringe the '712 patent for numerous reasons, two of which are presented in this motion.[6] ASP neither utilizes "executable tags" nor "process[es] the source . . . [by] executing [executable] tags." In addition, DuFresne cannot rely on the doctrine of equivalents because no expert evidence has been offered on the subject and, by Court order, DuFresne cannot introduce

[6] Because there is no direct infringement, there is also no indirect infringement. *Dynacore Holdings Corp. v. U.S. Philips Corp.*, 363 F.3d 1263, 1272 (Fed.Cir.2004) ("Indirect infringement, whether inducement to infringe or contributory infringement, can only arise in the presence of direct infringement.").

such evidence. Moreover, even though DuFresne should not now be allowed to present a theory under the doctrine of equivalents, prosecution history estoppel would in any event bar that theory.

A. ASP does not infringe the '712 patent because ASP does not include "executable tags."

Executable tags are "***delimited*** computer codes that are ***replaced*** with static and/or dynamic values ***upon execution*** of the tag" (emphasis added). DuFresne alleges that the delimited code blocks in ASP files are executable tags. [Bush Decl., Ex. E at 5.] They are not. Nothing in either the ASP file or the intermediate file corresponds to such a tag.

The delimited code blocks in the ASP file are not "replaced" during generation of the intermediate file. [Fielding Decl., Ex. 1 at ¶ 23(e)(v); Ex. 4 at ¶ 10.] To the contrary, during the parsing that generates the intermediate file, the code blocks are directly copied into the intermediate file. [MUF at ¶ 2.] Moreover, the claimed replacement is required to occur "upon execution of the tag." ASP code blocks are not executed, but rather are copied, during parsing. [Fielding Decl., Ex. 1 at ¶ 39(a)(v); Ex. 4. at ¶ 10.]

Turning to the intermediate file, there too executable tags are absent. The ASP code block is not set off by delimiters in the intermediate file. [MUF at ¶ 3.] That file is merely a script composed of general-purpose computer codes similar to the prior art CGI scripts that DuFresne distinguished during prosecution. [Fielding Decl., Ex.1 at ¶ 39(a)(vii).] At least because the code blocks are not "delimited," they are not executable tags. [Fielding Decl., Ex. 4 at ¶¶ 10 & 12.]

B. ASP also does not infringe the '712 patent because "processing the source . . . by executing the tags" (claims 9 and 14) and "processing the source . . ., the process executing the tags" (claim 26) are absent in ASP.

The processing limitation in the asserted claims means "fully processing the hypertext source into which executable tags were inserted, ***by replacing*** each executable tag ***in the source*** with a corresponding value and expanding any resulting dynamic values until they are static" (emphasis added). DuFresne alleges that the ASP file is the source that is processed. [Bush Decl., Ex. E at 6.]

This limitation requires more than merely identifying a file as the "source" and identifying executable tags that are alleged to be replaced. The replacement must occur ***in the source where the tag is found***. [Fielding Decl., Ex. 1 at ¶ 39(b)(iii); Ex. 4 at ¶ 12.] This never happens with ASP. [Fielding Decl., Ex. 1 at ¶ 39(b)(iii); Ex. 4 at ¶ 12.] The code blocks within an ASP file are never replaced, let alone modified in away way. [Fielding Decl., Ex. 1 at ¶ 23(e)(v); Ex. 4 at ¶ 12.] They are simply copied to the intermediate file. [MUF at ¶ 2.]

Replacement occurs, if at all, subsequently, in going from the intermediate file to the HTML file. However, the intermediate file is not a "hypertext source" as required by the processing limitation. First, the intermediate file is not the file requested by the client computer.[7] [Fielding Decl., Ex. 1 at ¶ 23(a); Ex. 4 at ¶ 11.] The intermediate file also no longer contains "executable tags" (for reasons already described), which the claims require for the intermediate file to qualify as the hypertext source,[8] and also does not contain HTML tags "that indicate how a page should be displayed by a browser," [Fielding Decl., Ex. 1 at ¶ 23(e)(viii); Ex. 4 at ¶ 11.] as required by the Court's construction. Although the text "<html tag>" appears literally in the intermediate file, it is embedded within the general-purpose print command. [Fielding Decl., Ex. 1 at ¶ 23(e)(i).] If passed to a browser in this form, it would be meaningless. [*Id.* at ¶ 23(e)(viii).]

[7] Moreover, the file forwarded to the client is also not the same file into which the alleged executable tags were inserted (the ASP file).

[8] Asserted claims 9, 14 and 26 all require "executable tags in a hypertext source to a displayable page".

C. DuFresne cannot rely on the doctrine of equivalents because DuFresne has produced no expert evidence on the subject and has been precluded from doing so by the Court.

Reliance on the doctrine of equivalents requires particularized, linking testimony from an expert witness. *AquaTex Indus., Inc. v. Techniche Solutions*, 479 F.3d 1320, 1329 (Fed. Cir. 2007). David Klausner, DuFresne's expert, served an initial and a supplemental expert report on infringement according to the schedule set by the Court. In neither report did Mr. Klausner disclose any theory of infringement under the doctrine of equivalents.

More than fifteen months after the deadline for opening expert reports, Mr. Klausner served another expert report on infringement. The Court granted Microsoft's motion to strike that report, noting that the report was "filed well after the deadline for the filing of such reports set by the Court." [Order dated October 6, 2005.] In October 2005, DuFresne sought reconsideration of the Court's order. [Docket No. 228.] The Court referred the issue to the Discovery Master, and then adopted the report and recommendation of the Discovery Master finding that "there [was] no good reason why Mr. Klausner could not have opined on the doctrine of equivalents in the scheduled round of expert reports served in the fall of 2003." [Order dated June 16, 2006; Docket No. 262 at 15.] Microsoft understands the Court's recent allowance of supplementations to the expert reports to be limited to only those supplementations necessary to conform to the claim constructions, and does not affect the Court's earlier orders precluding Dufresne's reliance on the doctrine of equivalents altogether.

D. DuFresne additionally cannot rely on the doctrine of equivalents because of prosecution history estoppel.

Even though DuFresne should not now be allowed to present a theory under the doctrine of equivalents, prosecution history estoppel would in any event bar that theory. "[T]he doctrine of equivalents . . . is not a tool for expanding the protection of a patent after examination has

been completed. Thus, prosecution history estoppel limits the range of equivalents available to a patentee by preventing recapture of subject matter surrendered during prosecution of a patent." *Southwall Techs., Inc. v. Cardinal IG Co.*, 54 F.3d 1570, 1579 (Fed. Cir 1995) (internal citations omitted). "Prosecution history estoppel is one limitation on the scope of equivalents that a patentee can claim under the doctrine of equivalents, and it is a question of law." *Bayer AG*, 212 F.3d at 1251 (internal citations omitted). "Estoppel arises when an amendment is made to secure the patent and the amendment narrows the patent's scope." *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co., Ltd.*, 535 U.S. 722, 736 (2002). Where a patentee narrows a claim by amendment while prosecuting the patent, any narrowed limitation is presumed to have no equivalent at all. *Id.* at 739-40.

During prosecution of the '712 patent, DuFresne amended his claims to distinguish the Lyons reference cited by the Examiner. [Bush Decl., Ex. B at 3-6.] In claim 1, DuFresne added the limitation "forwarding the hypertext source processed at the server to the client." [Bush Decl., Ex. C at 2.] A similar limitation was also added to claim 26. [*Id.* at 4.] These limitations were added in order to require that the hypertext source into which the executable tags were inserted be the same file that is retrieved, processed, and forwarded to the client computer, and thus to distinguish the multiple-file approach used by Lyons to deliver web pages:

> "[T]he present invention allows dynamic web pages to be presented to users by having the server execute the tags within the source of a page before transmission to a client, in order to substitute the tag with data of various types." [Bush Decl., Ex. C at 9.]
>
> "There is no teaching or suggestion in Lyons of an HTML server processing a single standalone source having executable tags embedded within the script to include data into the page, before that script is sent to a client computer." [*Id*. at 11.]
>
> "No server-interpretation of executable tags within a single, self-sufficient page is taught or suggested by Lyons, or Kindel for that matter. That is, the present invention can operate on a single page…" [*Id*. at 12.]

> "The present invention provides a way to effectuate dynamic web pages while not having to change the source of the web pages at all and without requiring user or client interaction. Since in this invention, the server executes the executable tags, the information included into the page from tag execution need only change in order to provide customized web pages. No hypertext page reprogramming or alteration are needed…" [*Id*. at 13.]
>
> "Each of these independent claims recites that executable tags exist within a source of a page, and that the source of the page is processed at the server to execute the executable tags." [*Id.* at 14.]

Because this claim amendment was related to patentability, it precludes DuFresne from arguing that the multiple file approach utilized in ASP is an equivalent to the processing of the "hypertext source" in the asserted claims.

VI. CONCLUSION

For the reasons stated above, Microsoft's motion for summary judgment of non-infringement should be granted.

APPENDIX

TABLE 1		
Stage 1: The Original ASP File	Stage 2: The Intermediate File (after the parsing step)	Stage 3: The HTML File (after the processing step)
<html tag> <%[CODE BLOCK]%> <html tag>	printf("<html tag>") [CODE BLOCK] Printf("<html tag>")	<html tag> html tag(s) resulting from [CODE BLOCK] <html tag>

TABLE 2		
Stage 1: ASP File	Stage 2: Intermediate File (after the parsing step)	Stage 3: HTML File (after the processing step)
<html> <body> <% For i=1 To 5 printf" Microsoft " Next %> </body> </html>	printf("<html>") printf("<body>") For i=1 To 5 printf(" Microsoft ") Next printf("</body>") printf("</html>")	<html> <body> Microsoft Microsoft Microsoft Microsoft Microsoft </body> </html>

Claim Term	Claims	The Court's Tentative Construction
executable tag	9, 14, 26	Delimited computer codes that are replaced with static and/or dynamic values upon execution of the tag
processing the source … by executing the tags processing the source …, the process executing the executable tags	9, 14 26	Fully processing the hypertext source into which executable tags were inserted, by replacing each executable tag in the source with a corresponding value and expanding any resulting dynamic values until they are static.
hypertext source to a displayable page	9, 14, 26	Predefined text with codes which indicate how a page should be displayed by a browser

Dated May 25, 2007

/Jennifer K. Bush/

Frank E. Scherkenbach (BBO #653819)
Kurt L. Glitzenstein (BBO #565312)
Thomas A. Brown (BBO #657715)
Samuel E. Sherry (BBO #667527)
FISH & RICHARDSON P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070

Jennifer K. Bush (admitted *pro hac vice*)
Jason W. Wolff (admitted *pro hac vice*)
FISH & RICHARDSON P.C.
12390 El Camino Real
San Diego, CA 92130
Telephone: (858) 678-5070

Attorneys for Defendant
MICROSOFT CORPORATION

CERTIFICATE OF SERVICE

I hereby certify that MICROSOFT'S MEMORANDUM IN SUPPORT OF ITS MOTION FOR SUMMARY JUDGMENT OF NON-INFRINGEMENT has been filed through the ECF system and will be sent electronically to the registered participants as identified on the Notice of Electronic Filing (NEF) and paper copies will be sent to those indicated as non registered participants on May 25, 2007.

/s/ Jennifer K. Bush